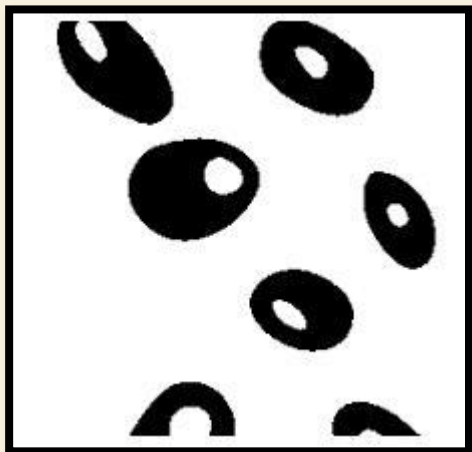


# **СИСТЕМЫ ТЕХНИЧЕСКОГО ЗРЕНИЯ**

ФИЛЬТРАЦИЯ В  
ПРОСТРАНСТВЕННОЙ  
ОБЛАСТИ.

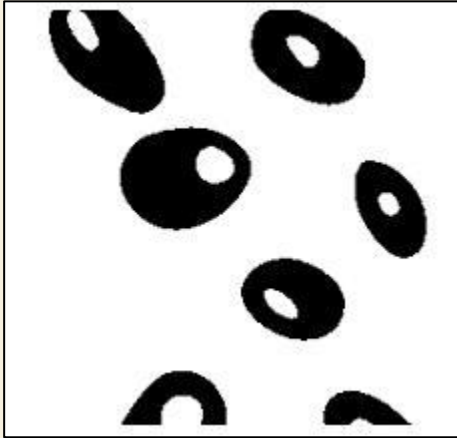
МОРФОЛОГИЧЕСКИЕ  
ОПЕРАЦИИ

# БИНАРНОЕ ИЗОБРАЖЕНИЕ

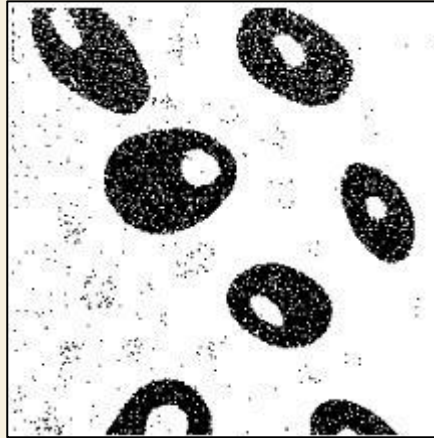


Модель шума «соль-перец»  
вероятности перехода

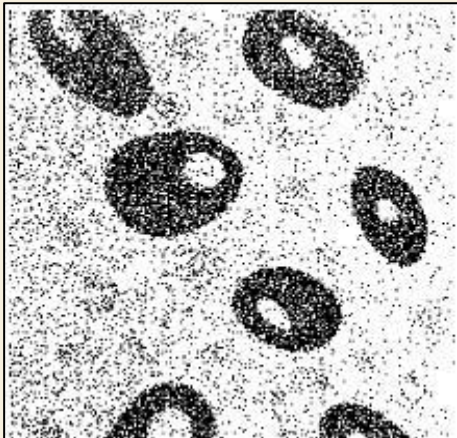
$Im[x,y] \rightarrow Im'[x,y]$	$Im'[x,y]=1$	$Im'[x,y]=0$
$Im[x,y]=1$	$1-p$	$p$
$Im[x,y]=0$	$q$	$1-q$



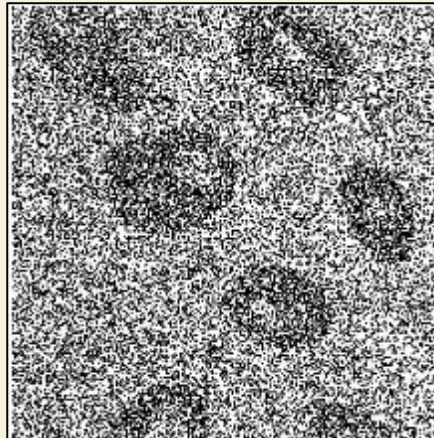
исходное



$p=0.1$   
 $q=0.1$



$p=0.25$   
 $q=0.25$



$p=0.45$   
 $q=0.45$

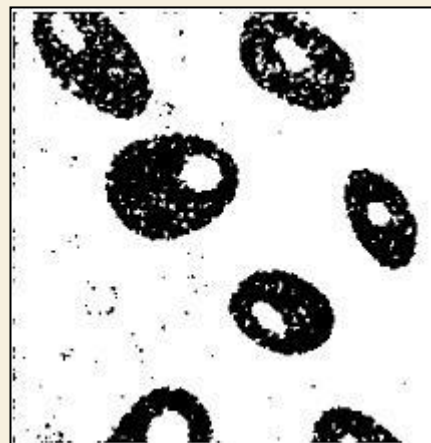
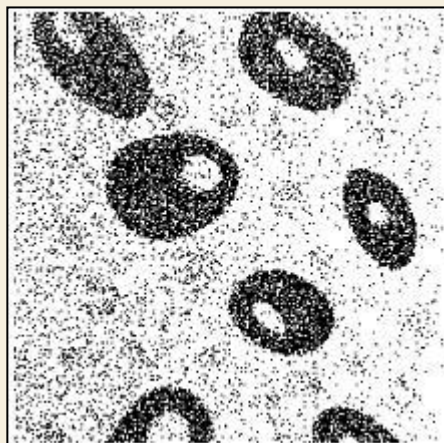
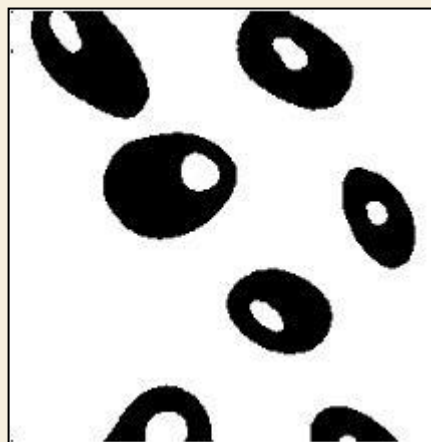
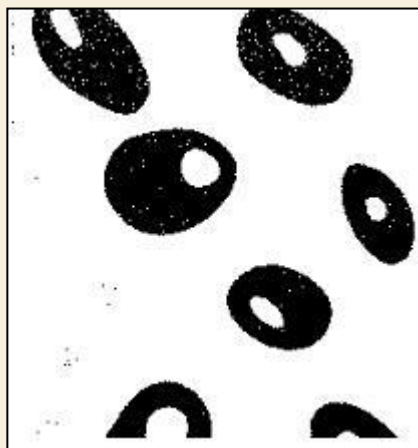
РАЗЛИЧНЫЙ  
УРОВЕНЬ ШУМА

# ФИЛЬТРАЦИЯ

- Медианный фильтр
- Считаем количество 0 и 1 в  $n$ -окрестности

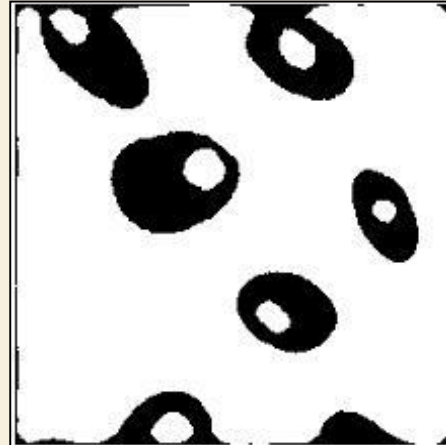
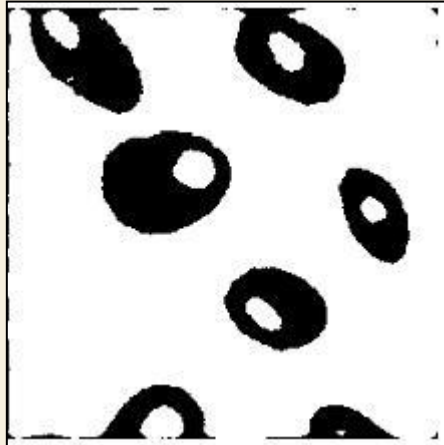
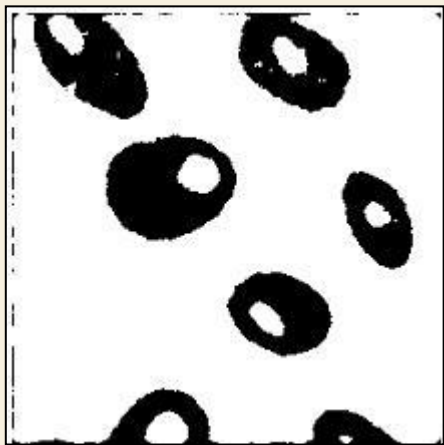
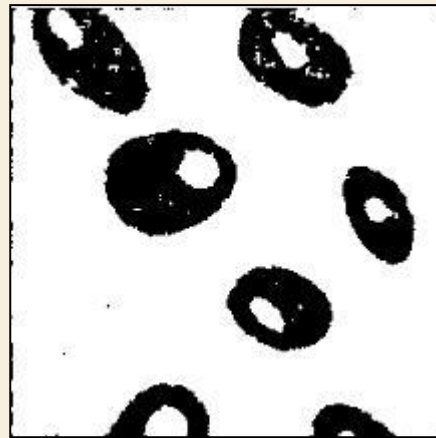
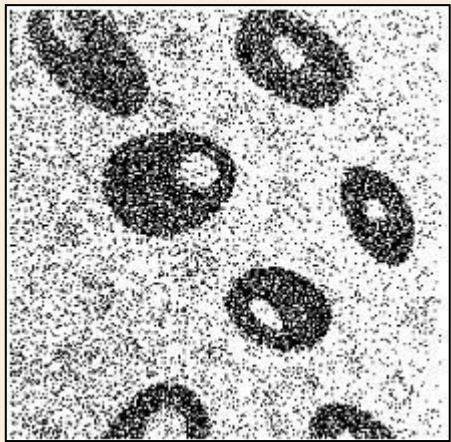
1	1	0
1	0	1
1	1	0

- Единиц > нулей  $\Rightarrow$  1



## БИНАРНОЕ ИЗОБРАЖЕНИЕ

Результат зависит от  
зашумлённости



## БИНАРНОЕ ИЗОБРАЖЕНИЕ

Результат зависит от  
размера апертуры

# РАНГОВЫЙ ФИЛЬТР

- ранговый фильтр – больше заданного порога
- разные пороги для нулей и единиц!
- Он же процентильный

1	1	0
1	0	1
1	1	0

- Единиц 6, нулей 3
- Ранг 6 – 1, ранг 7 – 0
- Нужен, если мы имеем априорную информацию

# ГАУССОВ ШУМ

- Модель аддитивного шума

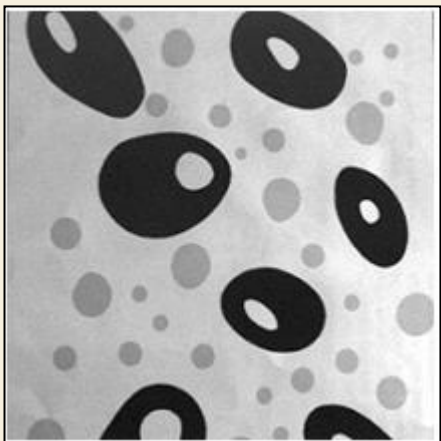
$$Im' [x, y] = Im[x, y] + R(x, y)$$

- Частный случай – гауссов шум

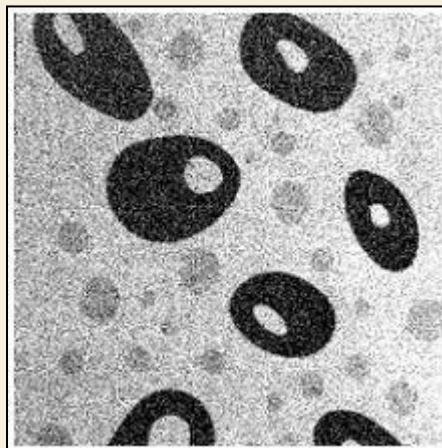
$$Im' [x, y] = Im[x, y] + N(0, \sigma),$$

- Мат.модель: сумма множества независимых факторов
- Подходит при маленьких дисперсиях
- Предположения: независимость, нулевое математическое ожидание

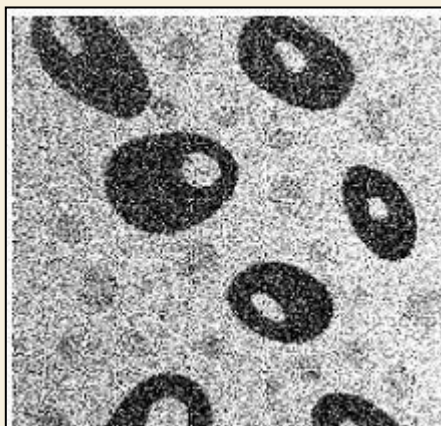




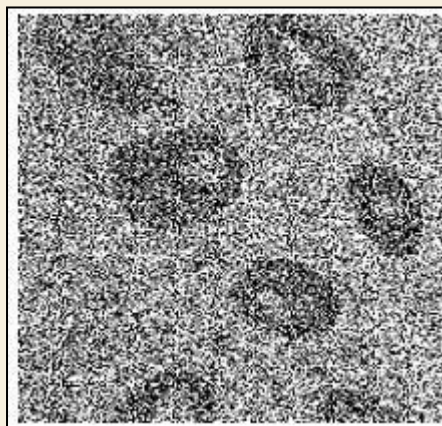
исходное



$\sigma=40$



$\sigma=80$



$\sigma=300$

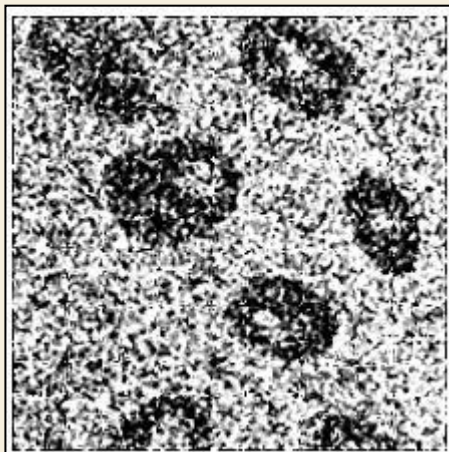
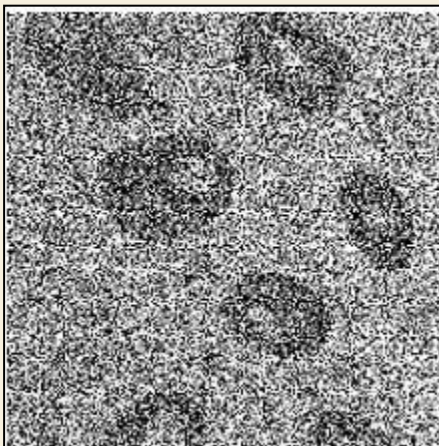
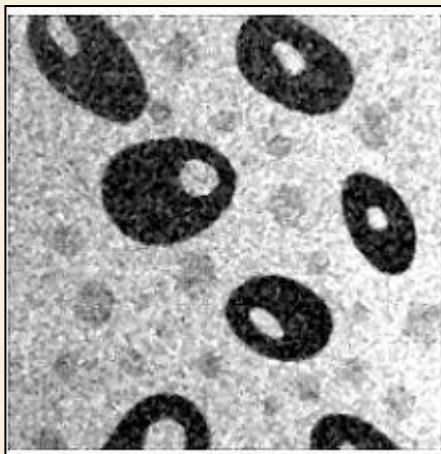
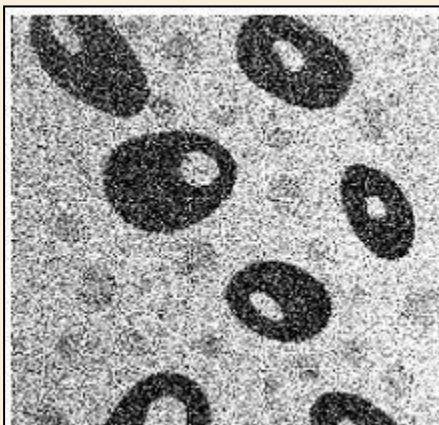
ГАУССОВ ШУМ

# МЕДИАННЫЙ ФИЛЬТР

- Почти как для бинарных изображений
- Упорядочиваем точки в  $n$ -окрестности

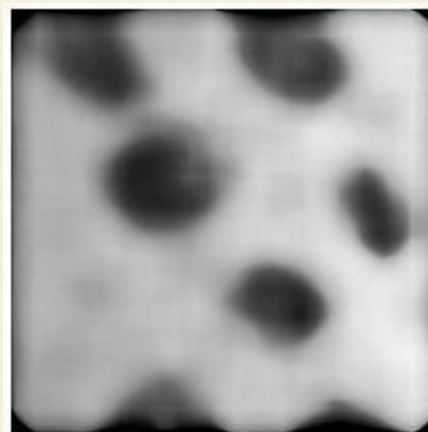
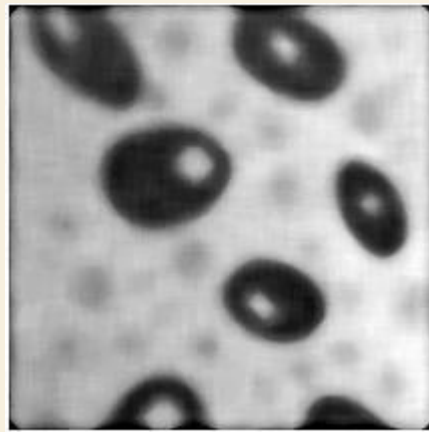
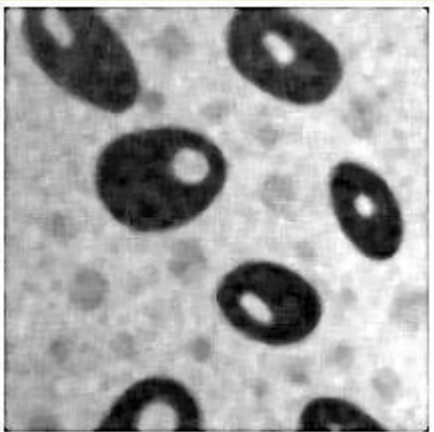
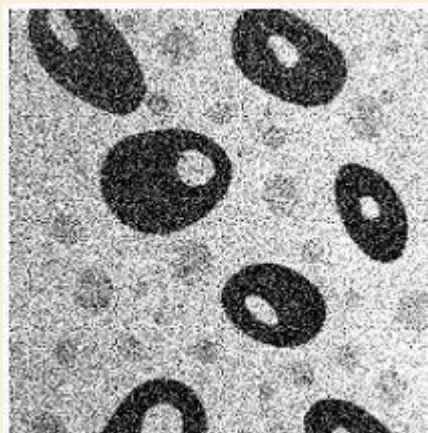
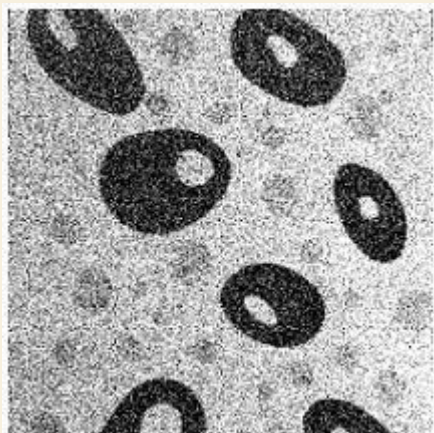
173	164	170
150	176	169
168	182	166

(150, 164, 166, 168, 169, 170, 173, 176, 182) — в  
середине списка



## ГАУССОВ ШУМ

Зависит от зашумлённости



## БИНАРНОЕ ИЗОБРАЖЕНИЕ

Зависит от размера апертуры

# РАНГОВЫЙ ФИЛЬТР

- Ранговый фильтр
- Упорядочиваем точки в  $n$ -окрестности

173	164	170
150	176	169
168	182	166

(150, 164, 166, 168, 169, 170, 173, 176, 182) –  
фильтр ранга 7

# СВЁРТКА С ПОСТОЯННЫМ ЯДРОМ

- Взвешенный фильтр
- Суммируем с постоянными весами  $W(x + a_i, y + b_i) = \text{const} = 1$

$$I(x, y) = \frac{1}{W_{\Sigma}} \sum_{x_i, y_i} I(x + x_i, y + y_i) W(x + x_i, y + y_i)$$

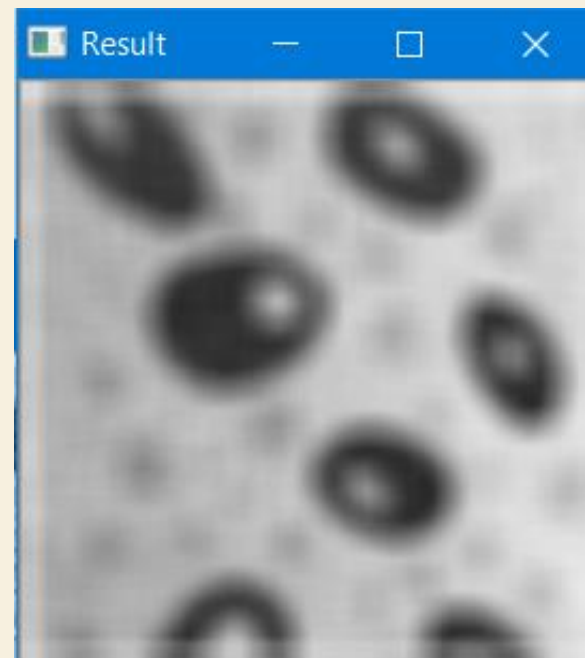
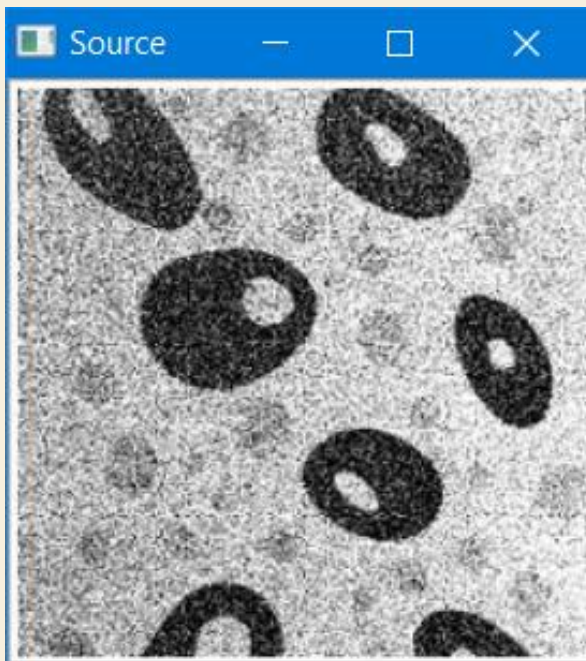
173	164	170
150	176	169
168	182	166

1	1	1
1	1	1
1	1	1

$\Sigma 167,33 \approx 167$



# СВЁРТКА С ПОСТОЯННЫМ ЯДРОМ



# СВЁРТКА С ФУНКЦИЕЙ ГАУССА

- Взвешенный фильтр

- Суммируем с весами  $W(x, y) = g(x, y) = e^{\frac{-x^2 - y^2}{2\sigma^2}}$

$$I(x, y) = \frac{1}{W_{\Sigma}} \sum_{a_i, b_i} I(x + a_i, y + b_i) W(x + a_i, y + b_i)$$

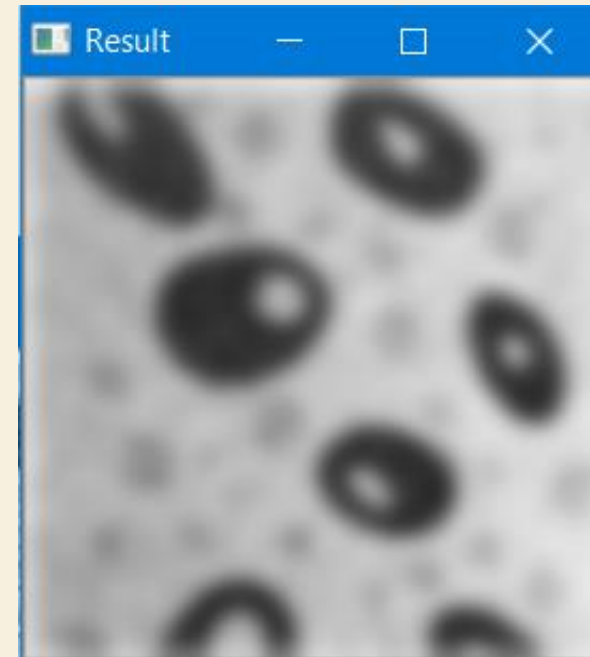
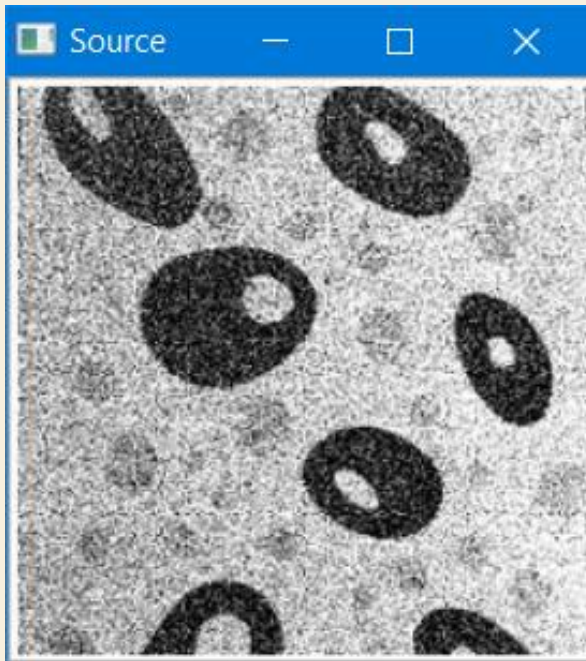
173	164	170
150	176	169
168	182	166

1	2	1
2	4	2
1	2	1

$$\Sigma 169,43 \approx 169$$



# СВЁРТКА С ФУНКЦИЕЙ ГАУССА



# БИЛАТЕРАЛЬНЫЙ ФИЛЬТР

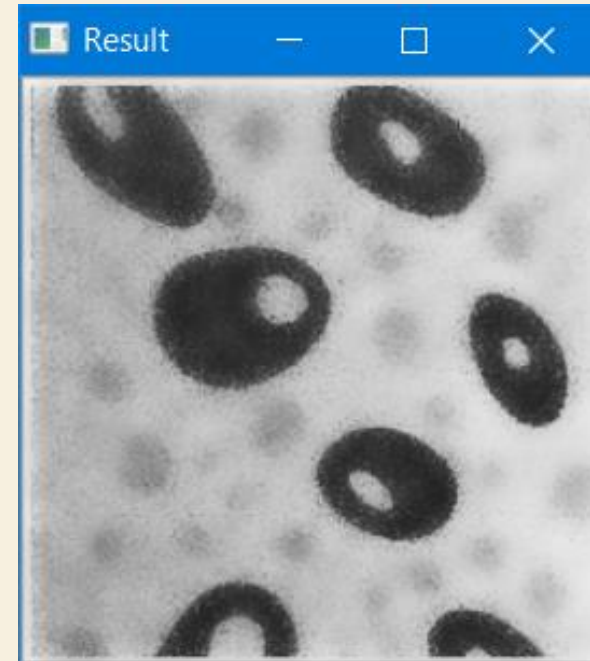
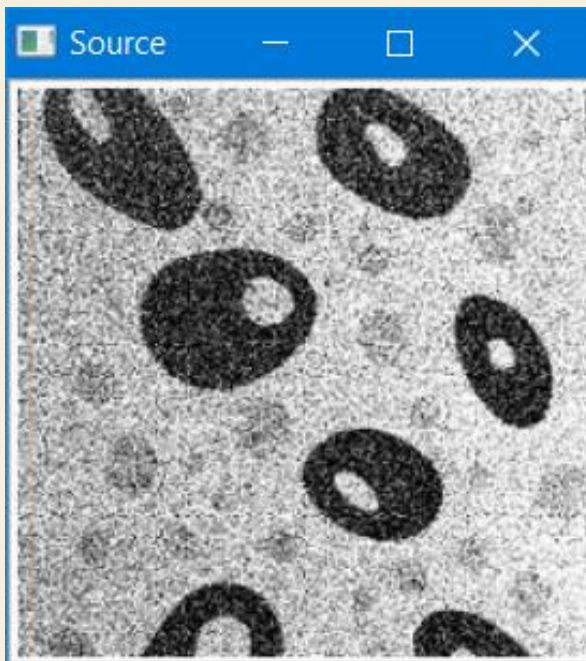
- Идея – мы будем свёртывать с функцией Гаусса, но одновременно с учётом яркости

$$r(I_i, I_0) = e^{\frac{-(I_i - I_0)^2}{2\sigma_r^2}}$$

$$I(x, y) = \frac{1}{W_\Sigma} \sum_{x_i, y_i} I(x + x_i, y + y_i) W(x + x_i, y + y_i) \cdot r(I(x, y), I(x + x_i, y + y_i))$$

- Такой фильтр лучше всего сохраняет края

# БИЛАТЕРАЛЬНЫЙ ФИЛЬТР



# ФИЛЬТРЫ В OPENCV

- Медианный фильтр

```
void medianBlur (cv::Mat src, CV::Mat dst, int ksize)
```

src – исходное изображение

dst – результат

ksize) – размер фильтра

- Бокс-фильтр

```
void boxFilter (cv::Mat src, cv::Mat dst, int depth, int ksize) // depth=-1
```

Он же blur()

- Гауссов фильтр

```
void gaussianBlur (cv::Mat src, cv::Mat dst, cv::Size (size_x size_y), sigma) // sigma = 0
```

- Билатеральный фильтр

```
bilateralFilter(src, dst, d, sigma_c, sigma_d); //d =0
```

# РЕЗУЛЬТАТ ФИЛЬТРАЦИИ

ОРИГИНАЛ



БОКС-ФИЛЬТР



# РЕЗУЛЬТАТ ФИЛЬТРАЦИИ

ОРИГИНАЛ



СВЁРТКА С ГАУССИАНОМ



# РЕЗУЛЬТАТ ФИЛЬТРАЦИИ

ОРИГИНАЛ



МЕДИАННЫЙ ФИЛЬТР



# РЕЗУЛЬТАТ ФИЛЬТРАЦИИ

ОРИГИНАЛ



БИЛАТЕРАЛЬНЫЙ ФИЛЬТР





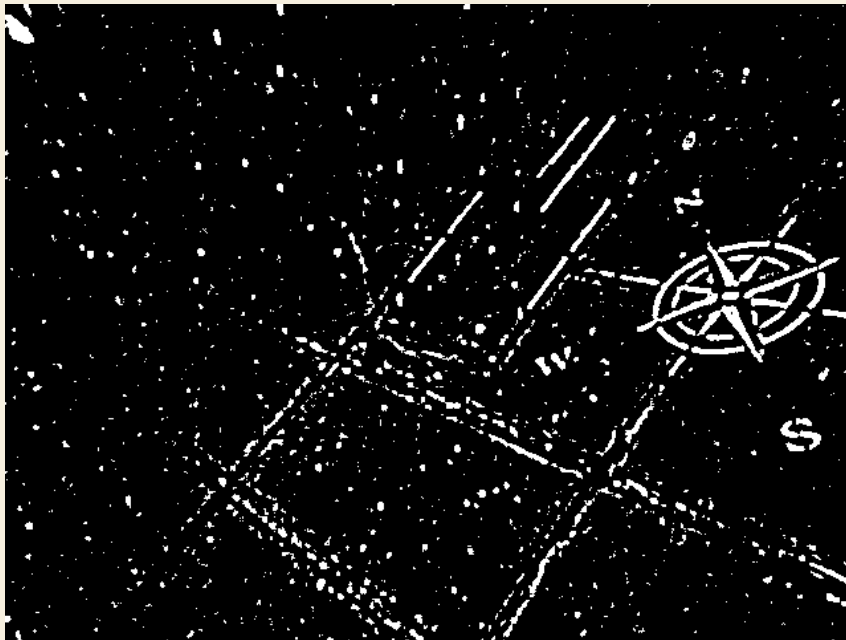
# МОРФОЛОГИЧЕСКИЕ ОПЕРАЦИИ

- Две морфологические операции – расширение и сжатие (dilate и erode)
- Сжатие – ранговый фильтр минимального ранга
- Расширение - максимального

1	1	0
1	0	1
1	1	0

- В данном случае минимальный ранг (нулевой) – 0
- Максимальный (девятый) ранг - 1

# МОРФОЛОГИЧЕСКИЕ ОПЕРАЦИИ



# МОРФОЛОГИЧЕСКИЕ ОПЕРАЦИИ

- Сочетание операций сужения и расширения – операция открытия – позволяет убирать одиночные «белые» пикселы
- Можно в обратном порядке – получим операцию закрытия

1	1	0
1	0	1
1	1	0

# МОРФОЛОГИЧЕСКИЕ ОПЕРАЦИИ

- Как уменьшить толщину контура?
- Построение скелета объектов
- Простейший вариант – итерационное применение морфологических операций
- Выберем крестообразный шаблон

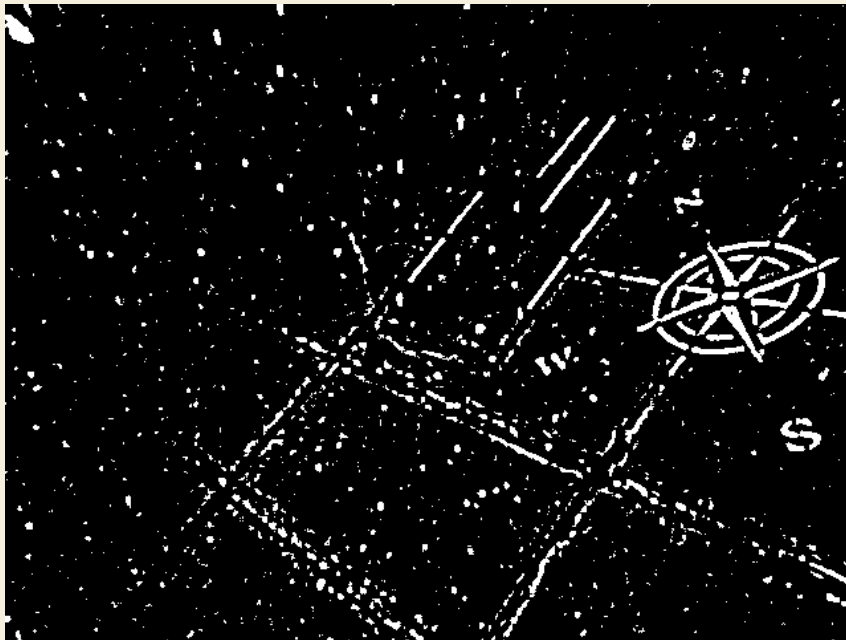
1	1	0
1	0	1
1	1	0

# МОРФОЛОГИЧЕСКИЕ ОПЕРАЦИИ

```
cv::Mat thinner (cv::Mat& source)
{
    cv::Mat element = cv::getStructuringElement(cv::MORPH_CROSS, cv::Size(3,3)), eroded, dilated;
    cv::Mat1b ret(source.size()), temp(source.size());
    int count = 0, done = 0;

    do
    {
        cv::erode(source, eroded, element);
        cv::dilate(eroded, dilated, element);
        cv::subtract(source, dilated, temp);
        cv::bitwise_or(ret, temp, ret);
        eroded.copyTo(source);
        done = cv::countNonZero(eroded);
        count++;
    }
    while ((count<10)&&(done));
    return ret;
}
```

# МОРФОЛОГИЧЕСКИЕ ОПЕРАЦИИ



Opencv\_morphology